

面目全非，胸有丘壑

人若无名，便可专心练剑

☰ 目录视图

☰ 摘要视图

RSS 订阅

个人资料



CouchDB

访问： 150922次
积分： 1810
等级： 4
排名： 第16887名

原创： 84篇
转载： 0篇
译文： 0篇
评论： 101条

文章分类

- Algorithm (22)
- Deep Learning (0)
- Parallel (0)
- Data Mining (2)
- Android (7)
- C++ (15)
- STL (14)
- Linux C (4)
- Java (0)
- 心情 (7)
- 瞎搞 (18)
- DirectX (8)

文章存档

- 2016年08月 (1)
- 2016年04月 (3)
- 2016年03月 (5)
- 2016年02月 (1)
- 2016年01月 (4)

展开

阅读排行

- 内存映射文件 (70994)
- Windows 多进程通信API (20331)

微信小程序实战项目——点餐系统 程序员11月书讯，评论得书啦 Get IT技能知识库，50个领域一键直达

C++ String类的构造函数、拷贝构造函数的实现

标签： 构造函数 析构函数 赋值函数 位拷贝 值拷贝

2015-10-10 10:51 419人阅读 评论

☰ 分类： C++ (14) ▾

❗ 版权声明：本文为博主原创文章，未经博主允许不得转载。

构造函数、析构函数与赋值函数是每个类最基本的函数，在一些公司的面试中也会经常问到这方面的问题。每个类只有一个析构函数和一个赋值函数，但可以有多个构造函数（包含一个拷贝构造函数，其它的称为普通构造函数）。对于任意一个类A，如果不手动编写上述函数，C++编译器将自动为类A生成四个缺省的函数：

```
A(void);           // 缺省的无参数构造函数

A(const A &a);       // 缺省的拷贝构造函数

~A(void);           // 缺省的析构函数

A& operate =(const A &a); // 缺省的赋值函数
```

虽然有自动生成，但是还是有必要手动写上述函数的。因为：

- （1）如果使用“缺省的无参数构造函数”和“缺省的析构函数”，等于放弃了自主“初始化”和“清除”的机会，C++发明人Stroustrup的好心好意白费了。
- （2）“缺省的拷贝构造函数”和“缺省的赋值函数”均采用“位拷贝”而非“值拷贝”的方式来实现，倘若类中含有指针变量，这两个函数注定将出错。

下面以类String的设计与实现为例，深入探讨这个道理。String的结构如下：

```
[cpp] view plain copy print ?
01. class String{
02. private:
03.     char *m_data;//成员变量，用于保存字符串
04.
05. public:
06.     String(const char *str=NULL);//普通构造函数
07.     String(const String &other);//拷贝构造函数
08.     ~String();//析构函数
09.     String &operator=(const String &other);//赋值函数
10. };
```

String类的普通构造函数和析构函数实现如下：

```
[cpp] view plain copy print ?
01. //String的普通构造函数
```

IFTTT的数据架构	(4574)
REST实战——调用百度	(4200)
map和multimap映射容器	(3842)
用Xposed框架拦截微信、	(3522)
基于MFC的FTP客户端	(1929)
2016阿里校招算法工程师	(1880)
STL 堆排序	(1788)
Windows多线程问题	(1583)

评论排行

用Xposed框架拦截微信、	(36)
Android操作系统MMS保	(13)
基于xposed的短信拦截	(11)
Windows 多进程通信API	(7)
C++虚函数的底层实现原	(6)
STL 堆排序	(4)
Intel HEX文件解析	(4)
map和multimap映射容器	(4)
Windows多线程问题	(2)
关于矩形排样问题(三)	(2)

最新评论

关于矩形排样问题(一)
qq_35806162: 大神，可不可以分享一下排料的源码

用Xposed框架拦截微信、人人、a134154: @slimboy123:请问你做好了吗？我定位失败了

用Xposed框架拦截微信、人人、CouchDB: @slimboy123:做这个的目的是什么？

用Xposed框架拦截微信、人人、slimboy123: @u011000290:模拟微信定位（附近的人）

用Xposed框架拦截微信、人人、CouchDB: @slimboy123:不知道。。。你是做什么项目，需要篡改系统的GPS？

用Xposed框架拦截微信、人人、slimboy123: @u011000290:楼主知不知道还有没有别的方式能实现？开启系统的允许模拟位置这种也没有作用， ...

用Xposed框架拦截微信、人人、CouchDB: @slimboy123:这个方案，当时是在TCL S838M真机上演示的，没有任何问题。不过Xpose...

用Xposed框架拦截微信、人人、slimboy123: @u011000290:最近才有这方面的项目需求

用Xposed框架拦截微信、人人、CouchDB: @slimboy123:。。。很久远的东西了，怎么这么多人还在问

用Xposed框架拦截微信、人人、slimboy123: 没什么效果，也不报错。清理内存、位置能搞的都搞了，不起作用，微信里面的定位依然没有改变

推荐文章

* 程序员10月书讯，评论得书

* Android中Xposed框架篇---修改系统位置信息实现自身隐藏功能

* Chromium插件（Plugin）模块（Module）加载过程分析

* Android TV开发总结--构建一个

```
02. String::String(const char *str)
03. {
04.     if (str==NULL)
05.     {
06.         m_data=new char[1];
07.         *m_data='\0';
08.     }
09.     else
10.     {
11.         int length=strlen(str);
12.         m_data=new char[length+1];
13.         strcpy(m_data,str);
14.     }
15. }
16. //String类的析构函数
17. String::~String(void)
18. {
19.     delete [] m_data;
20. }
```

刚刚上面说，如果不主动编写拷贝构造函数和赋值函数，编译器将以“位拷贝”的方式自动生成缺省的函数。如果String类中含有指针变量，那么这两个缺省的函数就隐含了错误。以类String的两个对象a、b为例，假设a.m_data的内容为“hello”，b.m_data的内容为“world”。位拷贝拷贝的是地址，而值拷贝则拷贝的是内容。现将a赋给b，缺省赋值函数的“位拷贝”意味着执行b.m_data = a.m_data，虽然b.m_data所指向的内容会变成“hello”，但是这将造成三个错误：一是b.m_data原有的内存没被释放，造成内存泄露；二是b.m_data和a.m_data指向同一块内存，a或b任何一方变动都会影响另一方；三是在对象被析构时，m_data被释放了两次。

对于编译器，如果不主动编写拷贝函数和赋值函数，它会以“位拷贝”的方式自动生成缺省的函数。如果重写赋值函数和拷贝构造函数后，b.m_data=a.m_data，进行的是值拷贝，会将a.m_data的内容赋给b.m_data,b.m_data还是指向原来的内存区域，但是其内容改变。

有下面4个语句：

String a(“hello”);

String b(“world”);

String c = a; // 调用了拷贝构造函数，最好写成 c(a);

c = b; // 调用了赋值函数

第3语句的风格较差，宜改写成String c(a) 以区别于第4语句

下面是类String的拷贝构造函数与赋值函数

```
【cpp】view plain copy print ?
01. // 拷贝构造函数
02. String::String(const String &other)
03. {
04.     // 允许操作other的私有成员m_data
05.     int length=strlen(other.m_data);
06.     m_data=new char[length+1];
07.     strcpy(m_data,other.m_data);
08. }
09.
10. //赋值函数
11. String & String::operator = (const String &other)
12. {
13.     //检查自赋值
14.     if (this==&other)
15.     {
16.         return *this;
17.     }
18.     //释放原有的内存资源
19.     delete []m_data;
20.     //分配新的内存资源，并复制内容
21.     int length=strlen(other.m_data);
22.     m_data=new char[length+1];
23.     strcpy(m_data,other.m_data);
```

TV app的直播节目实例

* 架构设计：系统存储--MySQL
简单主从方案及暴露的问题

文章搜索

Q

```
24. | //返回本对象的引用
25. | return *this;
26. | }
```

类String拷贝构造函数与普通构造函数（参见9.4节）的区别是：在函数入口处无需与NULL进行比较，这是因为“引用”不可能是NULL，而“指针”可以为NULL。

类String的赋值函数比构造函数复杂得多，分四步实现：

- (1) 第一步，检查自赋值。
- (2) 第二步，用delete释放原有的内存资源。如果现在不释放，以后就没机会了，将造成内存泄露。
- (3) 第三步，分配新的内存资源，并复制字符串。
- (4) 第四步，返回本对象的引用，目的是为了实现象 `a = b = c` 这样的链式表达。注意不要将 `return *this` 错写成 `return this`。

Ref

高质量C++/C编程指南

http://man.chinaunix.net/develop/c&c++/c/c.htm#_Toc520634051

顶 踩

1 0

上一篇 [Linux流编程](#)

下一篇 [strlen、strcpy、strcat等字符串处理函数的实现](#)

我的同类文章

C++ (14)

• REST实战——调用百度语...

2016-04-19

阅读 4200

• 利用lex和yacc做词法、语法...

2016-03-25

阅读 279

• Arduino通信协议设计

2016-03-02

阅读 363

• 深拷贝和浅拷贝

2015-12-30

阅读 200

• strlen、strcpy、strcat等字...

2015-10-17

阅读 291

• Intel HEX文件解析

2016-04-05

阅读 915

• Arduino自定义通信协议解析

2016-03-04

阅读 585

• C++虚函数的底层实现原理

2016-01-11

阅读 488

• 基于OK6410嵌入式WinCE ...

2015-12-21

阅读 304

• 内存映射文件

2015-08-28

阅读 71027

更多文章

猜你在找

- C++面试笔试精要
- c++面向对象前言及意见征集（来者不拒）视频课程
- Linux环境C++编程基础视频课程
- C++语言基础
- C++基础第一季
- C++什么样的类需要编写拷贝构造函数
- C++类中拷贝构造函数详解
- C++ Primer 学习笔记_23_类与数据抽象5_初始化列表
- C++类中拷贝构造函数详解
- c++类中系统默认拷贝构造函数的解析

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点, 不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved

